

**AN  
INTERNSHIP REPORT  
ON  
HUMAN AGE AND GENDER PREDICTION  
SYSTEM REPORT  
BY  
KAMAL ACHARYA  
(Tribhuvan University)**

**Date: 2023/12/12**

# Human Age and Gender Prediction

## ABSTRACT

In social interaction, gender and age play an important role. Natural differences vary in gender, as are the terms used to identify people by their age. Despite the importance of these factors in our daily lives, the machine's capability to measure facial pictures reliably and effectively comes as part of what is required for industrial applications. For many years, automated facial recognition and gender and age estimation using Artificial Intelligence have attracted a lot of attention and have increased popularity. The use of a Deep Neural Network(DNN) to predict gender and age is suggested in this paper. Based on the widespread availability of facial images on the World Wide Web, mainly on social media, automatic facial recognition and prediction of gender and age using machine learning models have attracted a lot of consideration over a decade and have become increasingly popular. In all the existing literatures, gender is predicted as male or female.

# **CHAPTER-1**

## **1. INTRODUCTION**

### **1.1 ABOUT THE PROJECT**

Human face may be a storehouse of various information about including identity, gender, age, etc. the looks of face is affected considerably by aging. This plays a significant role in non-verbal communication between humans. Age and gender, two key facial attributes, play a really foundational role in social interactions, making age and gender estimation from one face image a very important task in machine learning applications, like access control, human-computer interaction, law enforcement, marketing intelligence and visual surveillance. Automatic gender classification and age detection may be a fundamental task in computer vision, which has recently attracted immense attention. It plays a very important role in an exceedingly wide selection of the real-world applications like targeted advertisement, forensic science, visual surveillance, content-based searching, human-computer interaction systems, etc. for instance we are able to use this method to display advertisement supported different gender and different age bracket. This method may be employed in different mobile applications where there's some age restricted content in order that only appropriate user can see this content.classify the gender as male or female Gender and Age is estimated with the assistance of Caffe Model Face detection the real time using opencv.

## **CHAPTER-2**

### **2. SYSTEM ANALYSIS**

#### **2.1 EXISTING SYSTEM**

Existing System is SVM-Support Vector Machine is utilized in defining the gender only

##### **2.1.1 DISADVANTAGES**

1. SVM Not suitable for large Dataset.
2. In training time SVM Process Slow.
3. Give Less Accuracy.

#### **2.2 PROPOSED SYSTEM**

In this project we using OpenCV to detect the faces. Open CV is one of the best method for detection process.DNN algorithm using find the gender and age.

##### **2.2.1 ADVANTAGES**

1. Saving time
2. Advanced security
3. Low cost
4. Open CV is free Source.

## **CHAPTER-3**

### **3.DESIGN AND DEVELOPMENT**

#### **3.1 DIGITAL IMAGE PROCESSING**

##### **INTRODUCTION**

The identification of objects in an image. This process would probably start with image processing techniques such as noise removal, followed by (low-level) feature extraction to locate lines, regions and possibly areas with certain textures.

The clever bit is to interpret collections of these shapes as single objects, e.g. cars on a road, boxes on a conveyor belt or cancerous cells on a microscope slide. One reason this is an AI problem is that an object can appear very different when viewed from different angles or under different lighting. Another problem is deciding what features belong to what object and which are background or shadows etc. The human visual system performs these tasks mostly unconsciously but a computer requires skillful programming and lots of processing power to approach human performance. Manipulating data in the form of an image through several possible techniques. An image is usually interpreted as a two-dimensional array of brightness values, and is most familiarly represented by such patterns as those of a photographic print, slide, television screen, or movie screen. An image can be processed optically or digitally with a computer.

## **CLASSIFICATION OF IMAGES:**

There are 3 types of images used in Digital Image Processing. They are

- Binary Image
- Gray Scale Image
- Color Image

### **BINARY IMAGE:**

A binary image is a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.

Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit (0 or 1). This name black and white, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as grayscale images.

Binary images often arise in digital image processing as masks or as the result of certain operations such as segmentation, thresholding, and dithering. Some input/output devices, such as laser printers, fax machines, and bi-level computer displays, can only handle bi-level images.

## **GRAY SCALE IMAGE**

A grayscale Image is digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray(0-255), varying from black(0) at the weakest intensity to white(255) at the strongest.

Grayscale images are distinct from one-bit black-and-white images, which in the context of computer imaging are images with only the two colors, black, and white (also called bi-level or binary images). Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation.

Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum (e.g. infrared, visible light, ultraviolet, etc.), and in such cases they are monochromatic proper when only a given frequency is captured. But also they can be synthesized from a full color image; see the section about converting to grayscale.

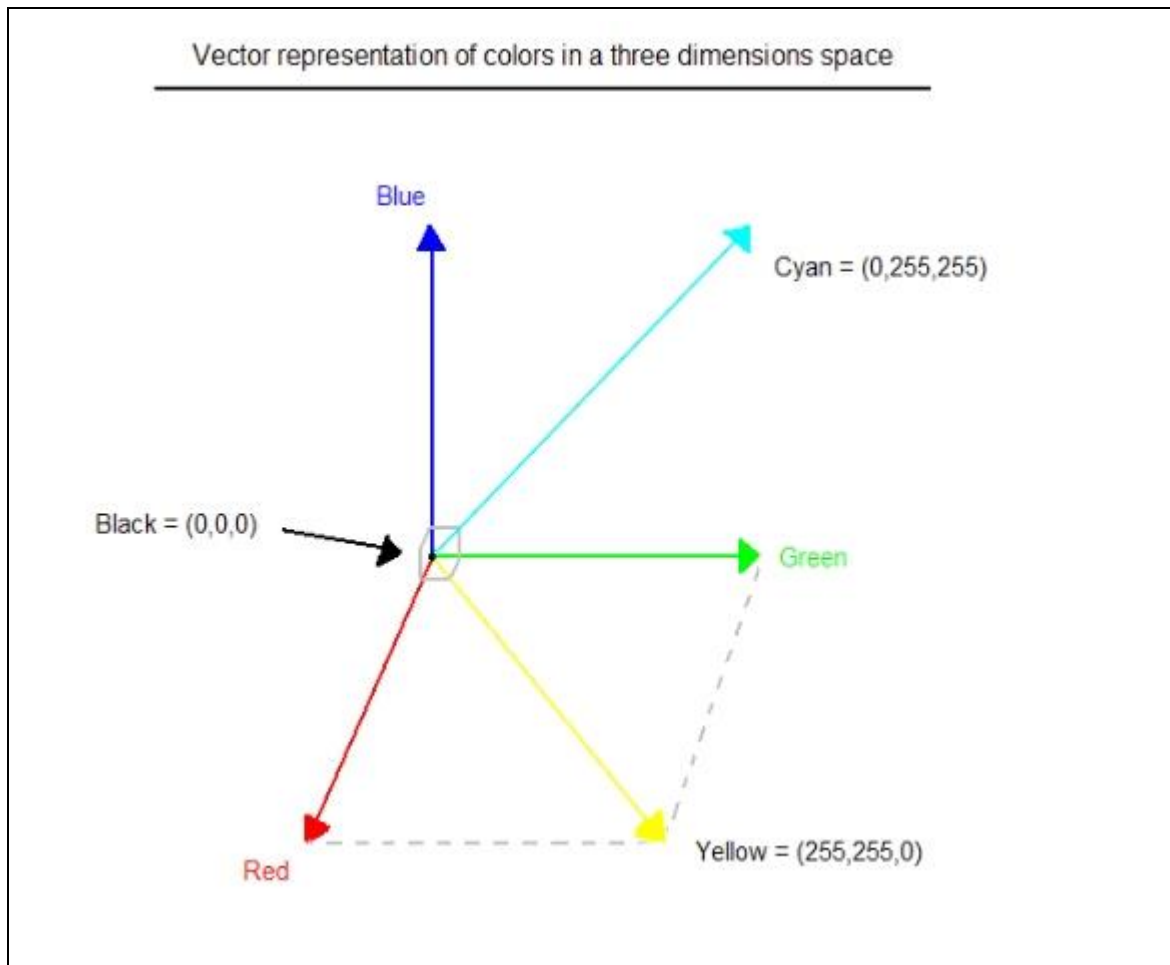
## **COLOUR IMAGE:**

A (digital) color image is a digital image that includes color information for each pixel. Each pixel has a particular value which determines its appearing color. This value is qualified by three numbers

giving the decomposition of the color in the three primary colors Red, Green and Blue. Any color visible to human eye can be represented this way. The decomposition of a color in the three primary colors is quantified by a number between 0 and 255. For example, white will be coded as  $R = 255, G = 255, B = 255$ ; black will be known as  $(R,G,B) = (0,0,0)$ ; and say, bright pink will be :  $(255,0,255)$ .

In other words, an image is an enormous two-dimensional array of color values, pixels, each of them coded on 3 bytes, representing the three primary colors. This allows the image to contain a total of  $256 \times 256 \times 256 = 16.8$  million different colors. This technique is also known as RGB encoding, and is specifically adapted to human vision





**Fig.1**

From the above figure, colors are coded on three bytes representing their decomposition on the three primary colors. It sounds obvious to a mathematician to immediately interpret colors as vectors in a three dimension space where each axis stands for one of the primary colors. Therefore we will benefit of most of the geometric mathematical concepts to deal with our colors, such as norms, scalar product, projection, rotation or distance.

# BASIC OF IMAGE PROCESSING

## 1.1 IMAGE

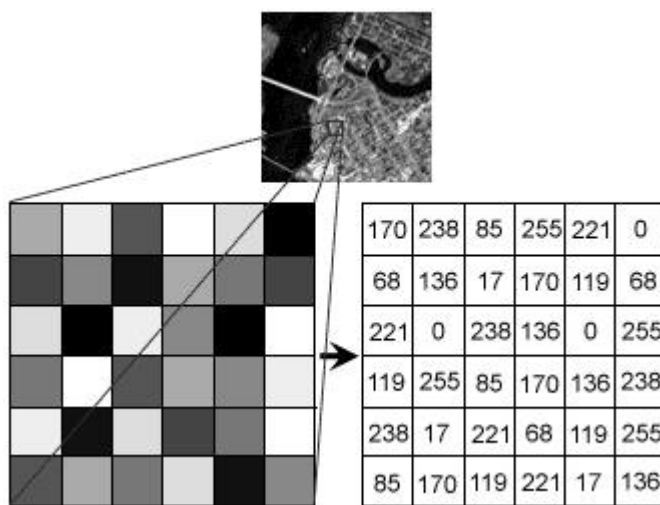
An image is a two-dimensional picture, which has a similar appearance to some subject usually a physical object or a person.

Image is a two-dimensional, such as a photograph, screen display, and as well as a three-dimensional, such as a statue. They may be captured by optical devices—such as cameras, mirrors, lenses, telescopes, microscopes, etc. and natural objects and phenomena, such as the human eye or water surfaces.

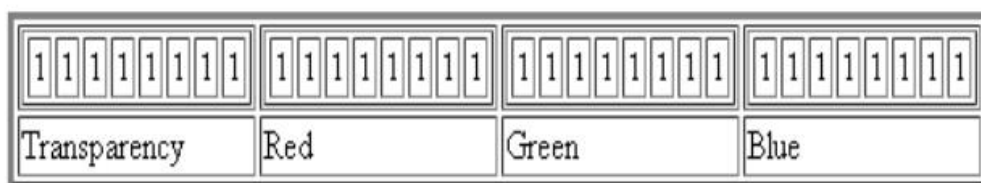
The word image is also used in the broader sense of any two-dimensional figure such as a map, a graph, a pie chart, or an abstract painting. In this wider sense, images can also be rendered manually, such as by drawing, painting, carving, rendered automatically by printing or computer graphics technology, or developed by a combination of methods, especially in a pseudo-photograph.



An image is a rectangular grid of pixels. It has a definite height and a definite width counted in pixels. Each pixel is square and has a fixed size on a given display. However different computer monitors may use different sized pixels. The pixels that constitute an image are ordered as a grid (columns and rows); each pixel consists of numbers representing magnitudes of brightness and color.



Each pixel has a color. The color is a 32-bit integer. The first eight bits determine the redness of the pixel, the next eight bits the greenness, the next eight bits the blueness, and the remaining eight bits the transparency of the pixel.



**Fig**

## **1.2 IMAGE FILE SIZES:**

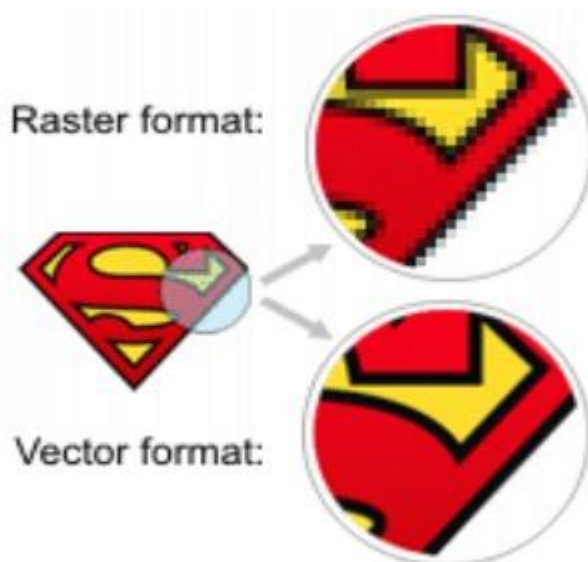
Image file size is expressed as the number of bytes that increases with the number of pixels composing an image, and the color depth of the pixels. The greater the number of rows and columns, the greater the image resolution, and the larger the file. Also, each pixel of an image increases in size when its color depth increases, an 8-bit pixel (1 byte) stores 256 colors, a 24-bit pixel (3 bytes) stores 16 million colors, the latter known as true color.

Image compression uses algorithms to decrease the size of a file. High resolution cameras produce large image files, ranging from hundreds of kilobytes to megabytes, per the camera's resolution and the image-storage format capacity. High resolution digital cameras record 12 megapixel (1MP = 1,000,000 pixels / 1 million) images, or more, in true color. For example, an image recorded by a 12 MP camera; since each pixel uses 3 bytes to record true color, the uncompressed image would occupy 36,000,000 bytes of memory, a great amount of digital storage for one image, given that cameras must record and store many images to be practical. Faced with large file sizes, both within the camera and a storage disc, image file formats were developed to store such large images.

### 1.3 IMAGE FILE FORMATS:

Image file formats are standardized means of organizing and storing images. This entry is about digital image formats used to store photographic and other images.

Image files are composed of either pixel or vector (geometric) data that are rasterized to pixels when displayed (with few exceptions) in a vector graphic display. Including proprietary types, there are hundreds of image file types. The PNG, JPEG, and GIF formats are most often used to display images on the Internet.



In addition to straight image formats, Metafile formats are portable formats which can include both raster and vector information. The metafile format is an intermediate format. Most Windows applications open metafiles and then save them in their own native format.

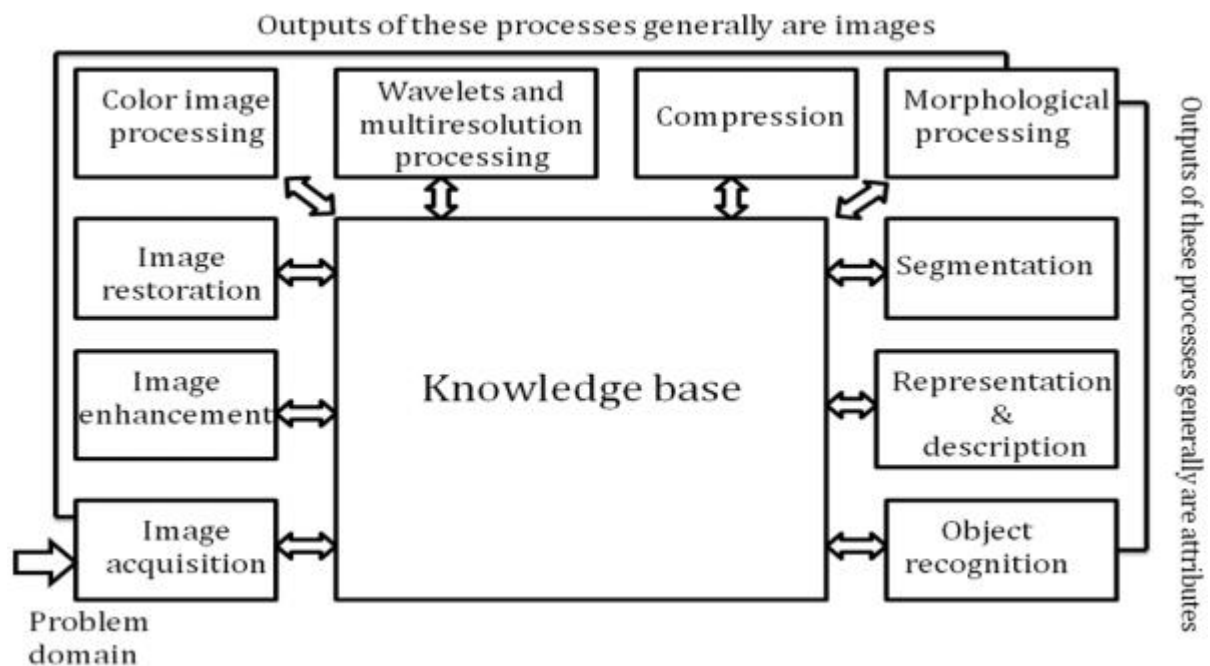
## **IMAGE PROCESSING:**

Digital image processing, the manipulation of images by computer, is relatively recent development in terms of man's ancient fascination with visual stimuli. In its short history, it has been applied to practically every type of images with varying degree of success.

The inherent subjective appeal of pictorial displays attracts perhaps a disproportionate amount of attention from the scientists and also from the layman. Digital image processing like other glamour fields, suffers from myths, misconnections, mis-understandings and mis-information. It is vast umbrella under which fall diverse aspect of optics, electronics, mathematics, photography graphics and computer technology. It is truly multidisciplinary endeavor ploughed with imprecise jargon.

Several factor combine to indicate a lively future for digital image processing. A major factor is the declining cost of computer equipment. Several new technological trends promise to further promote digital image processing. These include parallel processing mode practical by low cost microprocessors, and the use of charge coupled devices (CCDs) for digitizing, storage during processing and display and large low cost of image storage arrays.

## 3.2 FUNDAMENTAL STEPS IN DIGITAL IMAGE PROCESSING:



**Fig**

**Image Acquisition:** Image Acquisition is to acquire a digital image. To do so requires an image sensor and the capability to digitize the signal produced by the sensor. The sensor could be monochrome or color TV camera that produces an entire image of the problem domain every 1/30 sec. the image sensor could also be line scan camera that produces a single image line at a time. In this case, the objects motion past the line.

**Fig**

Scanner produces a two-dimensional image. If the output of the camera or other imaging sensor is not in digital form, an analog to digital converter digitizes it. The nature of the sensor and the image it produces are determined by the application.



**Fig**

### **Image Enhancement:**

**Image enhancement** is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interesting an image. A familiar example of enhancement is when we increase the contrast of an image because “it looks better.” It is important to keep in mind that enhancement is a very subjective area of image processing.



**Fig**



### 1.5.3 Image restoration:

**Image restoration** is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.



Enhancement, on the other hand, is based on human subjective preferences regarding what constitutes a “good” enhancement result. For example, contrast stretching is considered an enhancement technique because it is based primarily on the pleasing aspects it might present to the viewer, whereas removal of image blur by applying a deblurring function is considered a restoration technique.

### 3.3 Color image processing:

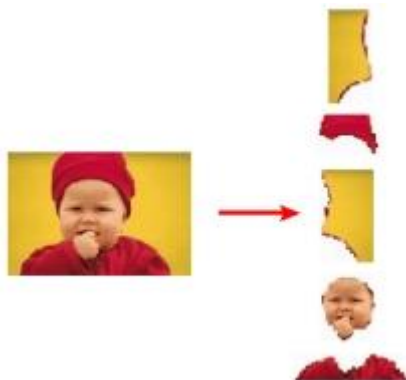
The use of color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades and intensities, compared to about only two dozen shades of gray. This second factor is particularly important in manual image analysis.



**Fig**

### **Segmentation:**

**Segmentation** procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.



**Fig**

On the other hand, weak or erratic segmentation algorithms almost always guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

Digital image is defined as a two dimensional function  $f(x, y)$ , where  $x$  and  $y$  are spatial (plane) coordinates, and the amplitude of  $f$  at any pair of coordinates  $(x, y)$  is called intensity or grey level of the image at that point. The field of digital image processing refers to processing digital images by means of a digital computer. The digital image is composed of a finite number of elements, each of which has a particular location and value. The elements are referred to as picture elements, image elements, pels, and pixels. Pixel is the term most widely used.

## **Image Compression**

Digital Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is removal of redundant data. From the mathematical viewpoint, this amounts to transforming a 2D pixel array into a statically uncorrelated data set. The data redundancy is not an abstract concept but a mathematically quantifiable entity. If  $n_1$  and  $n_2$  denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy  $R_D$  [2] of the first data set (the one characterized by  $n_1$ ) can be defined as,

$$R_D = 1 - \frac{1}{C_R}$$

Where  $C_R$  called as compression ratio [2]. It is defined as

$$C_r = \frac{n1}{n2}$$

In image compression, three basic data redundancies can be identified and exploited: Coding redundancy, interpixel redundancy, and psychovisual redundancy. Image compression is achieved when one or more of these redundancies are reduced or eliminated.

The image compression is mainly used for image transmission and storage. Image transmission applications are in broadcast television; remote sensing via satellite, air-craft, radar, or sonar; teleconferencing; computer communications; and facsimile transmission. Image storage is required most commonly for educational and business documents, medical images that arise in computer tomography (CT), magnetic resonance imaging (MRI) and digital radiology, motion pictures, satellite images, weather maps, geological surveys, and so on.

## Image Compression Model

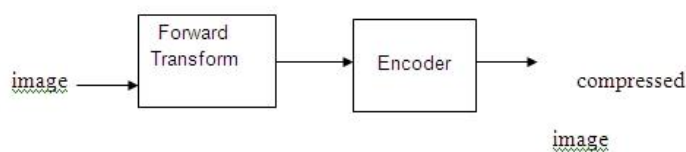
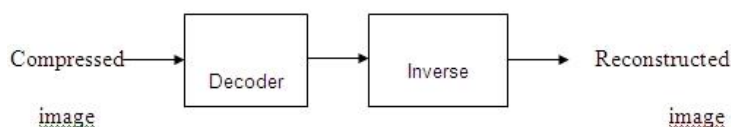


Figure 1.1.a) Block Diagram of Image compression



## Image Compression Types

There are two types' image compression techniques.

1. Lossy Image compression
2. Lossless Image compression

Compression ratio:

$$\text{compression ratio} = \frac{B_0}{B_1}$$

$B_0$  – number of bits before compression

$B_1$  – number of bits after compression

### 1. Lossy Image compression :

Lossy compression provides higher levels of data reduction but result in a less than perfect reproduction of the original image. It provides high compression ratio. lossy image compression is useful in applications such as broadcast television, videoconferencing, and facsimile transmission, in which a certain amount of error is an acceptable trade-off for increased compression performance.

Originally, PGF has been designed to quickly and progressively decode lossy compressed aerial images. A lossy compression mode has been preferred, because in an application like a terrain explorer texture data (e.g., aerial orthophotos) is usually mid-mapped filtered and therefore lossy mapped onto the terrain surface. In addition, decoding

lossy compressed images is usually faster than decoding lossless compressed images.

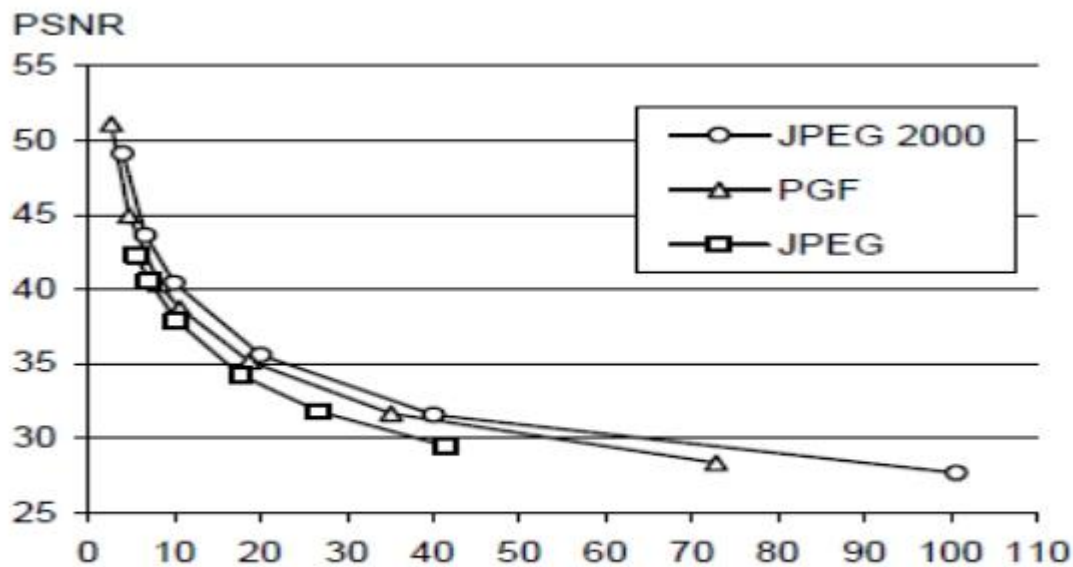


Fig. 4: PSNR of lossy compression in relation to compression ratio.

In the next test series we evaluate the lossy compression efficiency of PGF. One of the best competitors in this area is for sure JPEG 2000. Since JPEG 2000 has two different filters, we used the one with the better trade-off between compression efficiency and runtime. On our machine the 5/3 filter set has a better trade-off than the other. However, JPEG 2000 has in both cases a remarkable good compression efficiency for very high compression ratios but also a very poor encoding and decoding speed.

The other competitor is JPEG. JPEG is one of the most popular image file formats. It is very fast and has a reasonably good compression efficiency for a wide range of compression ratios. The

drawbacks of JPEG are the missing lossless compression and the often missing progressive decoding. Fig. 4 depicts the average rate-distortion behavior for the images in the Kodak test set when fixed (i.e., nonprogressive) lossy compression is used. The PSNR of PGF is on average 3% smaller than the PSNR of JPEG 2000, but 3% better than JPEG.

These results are also qualitative valid for our PGF test set and they are characteristic for aerial ortho-photos and natural images. Because of the design of PGF we already know that PGF does not reach the compression efficiency of JPEG 2000. However, we are interested in the trade-off between compression efficiency and runtime. To report this trade-off we show in Table 4 a comparison between JPEG 2000 and PGF and in Fig. 5 (on page 8) we show for the same test series as in Fig. 4 the corresponding average decoding times in relation to compression ratios.

Table 4 contains for seven different compression ratios (mean values over the compression ratios of the eight images of the Kodak test set) the corresponding average encoding and decoding times in relation to the average PSNR values. In case of PGF the encoding time is always slightly longer than the corresponding decoding time. The reason for that is that the actual encoding phase (cf. Subsection 2.4.2) takes slightly longer than the corresponding decoding phase.

For six of seven ratios the PSNR difference between JPEG 2000 and PGF is within 3% of the PSNR of JPEG 2000. Only in the first row is the difference larger (21%), but because a PSNR of 50 corresponds to an almost perfect image quality the large PSNR difference corresponds with an almost undiscoverable visual difference. The price they pay in JPEG 2000 for the 3% more PSNR is very high. The creation of a PGF is five to twenty times faster than the creation of a corresponding JPEG 2000 file, and the decoding of the created PGF is still five to ten times faster than the decoding of the JPEG 2000 file. This gain in speed is remarkable, especially in areas where time is more important than quality, maybe for instance in real-time computation.

Ratio	JPEG 2000 5/3			PGF		
	enc	dec	PSNR	enc	dec	PSNR
2.7	1.86	1.35	64.07	0.34	0.27	51.10
4.8	1.75	1.14	47.08	0.27	0.21	44.95
8.3	1.68	1.02	41.98	0.22	0.18	40.39
10.7	1.68	0.98	39.95	0.14	0.13	38.73
18.7	1.61	0.92	36.05	0.12	0.11	35.18
35.1	1.57	0.87	32.26	0.10	0.09	31.67
72.9	1.54	0.85	28.86	0.08	0.08	28.37

Table 4: Trade-off between quality and speed for the Kodak test set



In Fig. 5 we see that the price we pay in PGF for the 3% more PSNR than JPEG is low: for small compression ratios ( $< 9$ ) decoding in PGF takes two times longer than JPEG and for higher compression ratios ( $> 30$ ) it takes only ten percent longer than JPEG. These test results are characteristic for both natural images and aerial ortho-photos. Again, in the third test series we only use the ‘Lena’ image. We run our lossy coder with six different quantization parameters and measure the PSNR in relation to the resulting compression ratios. The results (ratio: PSNR) are:

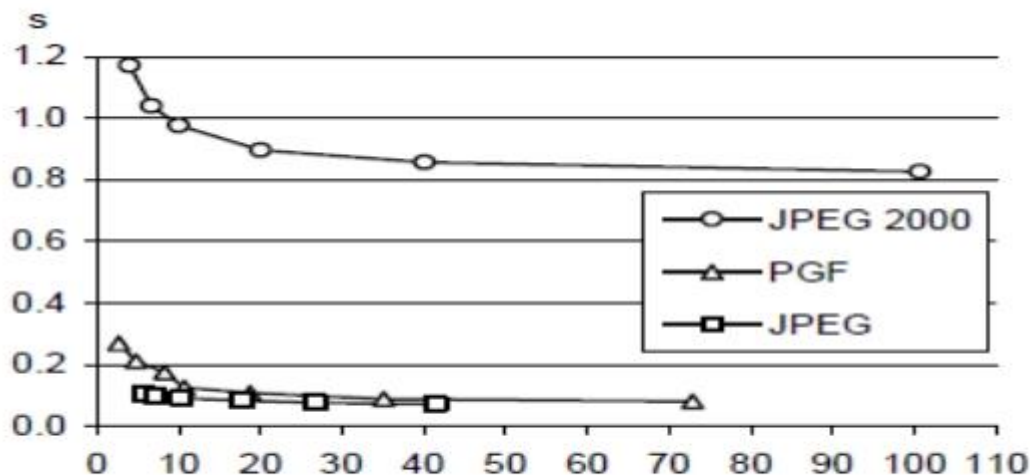


Fig. 5: Decoding time in relation to compression ratio

## 2. Lossless Image compression :

Lossless Image compression is the only acceptable amount of data reduction. It provides low compression ratio while compared to lossy. In Lossless Image compression techniques are composed of two

relatively independent operations: (1) devising an alternative representation of the image in which its interpixel redundancies are reduced and (2) coding the representation to eliminate coding redundancies.

Lossless Image compression is useful in applications such as medical imaginary, business documents and satellite images.

Table 2 summarizes the lossless compression efficiency and Table 3 the coding times of the PGF test set. For WinZip we only provide average runtime values, because of missing source code we have to use an interactive testing procedure with runtimes measured by hand. All other values are measured in batch mode

	WinZip	JPEG- LS	JPEG 2000	PNG	PGF
aerial	1.352	2.073	2.383	1.944	2.314
compound	12.451	6.802	6.068	13.292	4.885
hibiscus	1.816	2.200	2.822	2.087	2.538
houses	1.241	1.518	2.155	1.500	1.965
logo	47.128	16.280	12.959	50.676	10.302
redbrush	2.433	4.041	4.494	3.564	3.931
woman	1.577	1.920	2.564	1.858	2.556
average	9.71	4.98	4.78	10.70	4.07

Table 2: Lossless compression ratios of the PGF test set.

In Table 2 it can be seen that in almost all cases the best compression ratio is obtained by JPEG 2000, followed by PGF, JPEG-LS, and PNG. This result is different to the result in [SEA+00], where the best performance for a similar test set has been reported for JPEG-

LS. PGF performs between 0.5% (woman) and 21.3% (logo) worse than JPEG 2000. On average it is almost 15% worse. The two exceptions to the general trend are the ‘compound’ and the ‘logo’ images. Both images contain for the most part black text on a white background. For this type of images, JPEG-LS and in particular WinZip and PNG provide much larger compression ratios. However, in average PNG performs the best, which is also reported in [SEA+00].

These results show, that as far as lossless compression is concerned, PGF performs reasonably well on natural and aerial images. In specific types of images such as ‘compound’ and ‘logo’ PGF is outperformed by far in PNG.

	WinZip		JPEG-LS		JPEG 2000		PNG		PGF	
	enc	dec	enc	dec	enc	dec	enc	dec	enc	dec
a			1.11	0.80	5.31	4.87	3.70	0.19	0.99	0.77
c			1.61	0.38	3.46	3.06	2.95	0.18	0.95	0.80
hi			0.69	0.30	1.45	1.29	1.77	0.10	0.35	0.27
ho			0.65	0.30	1.62	1.47	0.85	0.11	0.41	0.32
l			0.09	0.02	0.26	0.21	0.16	0.01	0.07	0.06
r			0.65	0.44	4.29	4.01	3.61	0.16	0.66	0.59
w			0.39	0.30	1.76	1.63	1.08	0.08	0.35	0.27
av	1.14	0.37	0.74	0.36	2.59	2.36	2.02	0.12	0.54	0.44

Table 3: Runtime of lossless compression of the PGF test set

Table 3 shows the encoding (enc) and decoding (dec) times (measured in seconds) for the same algorithms and images as in Table 2. JPEG 2000 and PGF are both symmetric algorithms, while WinZip, JPEG-LS and in particular PNG are asymmetric with a clearly shorter decoding than encoding time. JPEG 2000, the slowest in encoding and decoding, takes more than four times longer than PGF. This speed gain is due to the simpler coding phase of PGF. JPEG-LS is slightly slower than PGF during encoding, but slightly faster in decoding images.

WinZip and PNG decode even more faster than JPEG-LS, but their encoding times are also worse. PGF seems to be the best compromise between encoding and decoding times. Our PGF test set clearly shows that PGF in lossless mode is best suited for natural images and aerial orthophotos. PGF is the only algorithm that encodes the three MByte large aerial ortho-photo in less than second without a real loss of compression efficiency. For this particular image the efficiency loss is less than three percent compared to the best. These results should be underlined with our second test set, the Kodak test set.

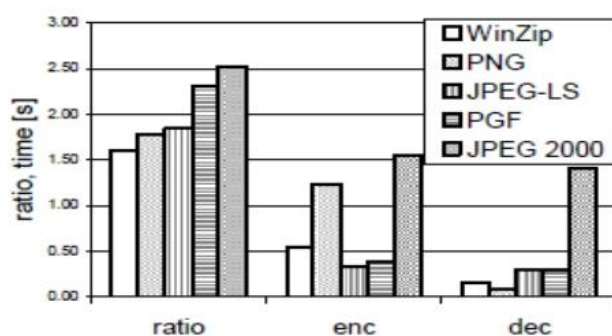


Fig. 3: Lossless compression results of the Kodak test set.

Fig. 3 shows the averages of the compression ratios (ratio), encoding (enc), and decoding (dec) times over all eight images. JPEG 2000 shows in this test set the best compression efficiency followed by PGF, JPEG-LS, PNG, and WinZip. In average PGF is eight percent worse than JPEG 2000. The fact that JPEG 2000 has a better lossless compression ratio than PGF does not surprise, because JPEG 2000 is more quality driven than PGF.

However, it is remarkable that PGF is clearly better than JPEG-LS (+21%) and PNG (+23%) for natural images. JPEG-LS shows in the Kodak test set also a symmetric encoding and decoding time behavior. Its encoding and decoding times are almost equal to PGF. Only PNG and WinZip can faster decode than PGF, but they also take longer than PGF to encode.

If both compression efficiency and runtime is important, then PGF is clearly the best of the tested algorithms for lossless compression of natural images and

aerial orthophotos. In the third test we perform our lossless coder on the ‘Lena’ image. The compression ratio is 1.68 and the encoding and decoding takes 0.25 and 0.19 seconds, respectively.

#### **1.4.4. Image Compression Standards**

There are many methods available for lossy and lossless, image compression. The efficiency of these coding standardized by some Organizations. The International Standardization Organization (ISO) and Consultative Committee of the International Telephone and

Telegraph (CCITT) are defined the image compression standards for both binary and continuous tone (monochrome and Colour) images. Some of the Image Compression Standards are

1. JBIG1
2. JBIG2
3. JPEG-LS
4. DCT based JPEG
5. Wavelet based JPEG2000

Currently, JPEG2000 [3] is widely used because; the JPEG-2000 standard supports lossy and lossless compression of single-component (e.g., grayscale) and multicomponent (e.g., color) imagery. In addition to this basic compression functionality, however, numerous other features are provided, including:

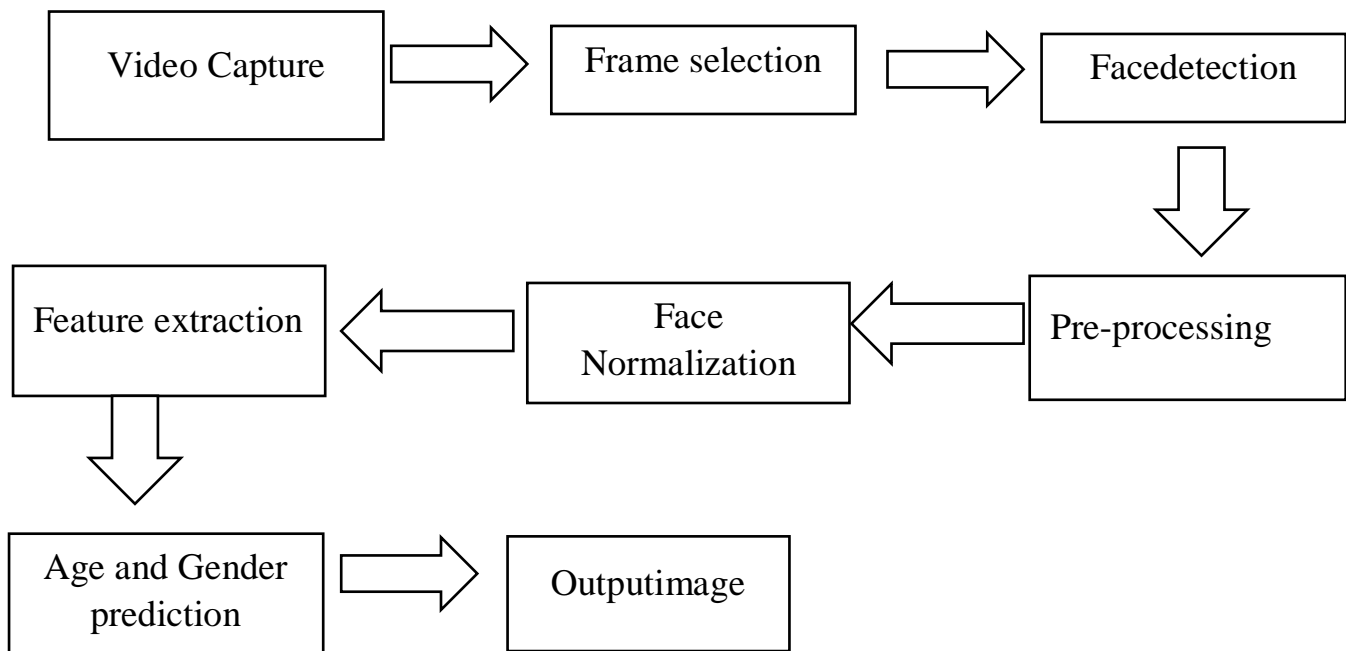
1) progressive recovery of an image by fidelity or resolution; 2) region of interest coding, whereby different parts of an image can be coded with differing fidelity; 3) random access to particular regions of an image without the need to decode the entire code stream; 4) a flexible file format with provisions for specifying opacity information and image sequences; and 5) good error resilience. Due to its excellent coding performance and many attractive features, JPEG 2000 has a very large potential application base. Some possible application areas include: image archiving, Internet, web browsing, document imaging,

digital photography, medical imaging, remote sensing, and desktop publishing.

The main advantage of JPEG2000 over other standards, First, it would addresses a number of weaknesses in the existing JPEG standard. Second, it would provide a number of new features not available in the JPEG standard.

The preceding points led to several key objectives for the new standard, namely that it should: 1) allow efficient lossy and lossless compression within a single unified coding framework, 2) provide superior image quality, both objectively and subjectively, at low bit rates, support additional features such as region of interest coding, and a more flexible file format, 4) avoid excessive computational and memory complexity. Undoubtedly, much of the success of the original JPEG standard can be attributed to its royalty-free nature. Consequently, considerable effort has been made to ensure that minimally-compliant JPEG- 2000 codec can be implemented free of royalties.

## BLCOK DIAGRAM



## OPEN CV

Open Computer Vision is shortly denoted as Open CV. Officially launched in 1999 the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team.

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. All of the new developments and algorithms appear in the C++ interface. There are bindings



in Python, Java and MATLAB. The API for these interfaces can be found in the online documentation. Wrappers in several programming languages have been developed to encourage adoption by a wider audience.

In our project Open Computer Vision used to detect the face in our face recognition technique. Open CV is act like eye of the computer or any machines.

## **CHAPTER-4**

### **4.1 CLASSIFICATION PROCESS**

Deep neural networks (DNNs), usually simply called neural networks (NNs), are computing systems inspired by the biological neural networks that constitute animal brains.

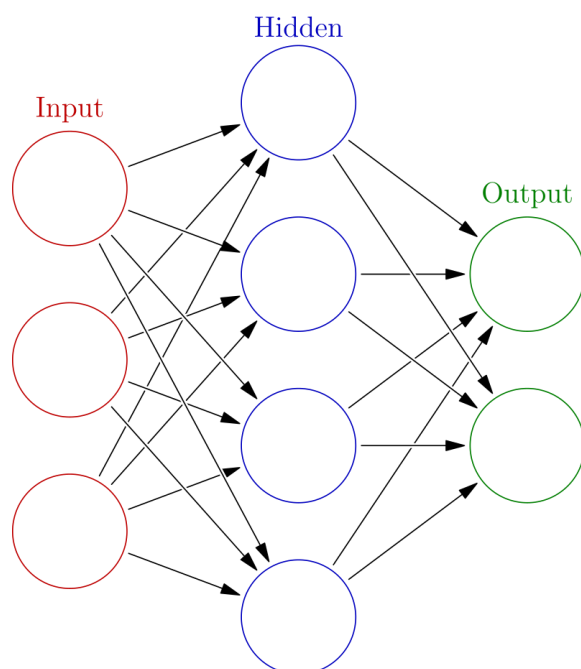
An DNN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from

the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

## 4.2 Components of DNNs

### Neurons

DNNs are composed of artificial neurons which are conceptually derived from biological neurons. Each artificial neuron has inputs and produces a single output which can be sent to multiple other neurons. The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons. The



outputs of the final output neurons of the neural net accomplish the task, such as recognizing an object in an image.

To find the output of the neuron, first we take the weighted sum of all the inputs, weighted by the weights of the connections from the inputs to the neuron. We add a bias term to this sum. This weighted sum is sometimes called the activation. This weighted sum is then passed through a (usually nonlinear) activation function to produce the output. The initial inputs are external data, such as images and documents. The ultimate outputs accomplish the task, such as recognizing an object in an image.

### **Connections and weights**

The network consists of connections, each connection providing the output of one neuron as an input to another neuron. Each connection is assigned a weight that represents its relative importance. A given neuron can have multiple input and output connections.

## **Propagation function**

The propagation function computes the input to a neuron from the outputs of its predecessor neurons and their connections as a weighted sum. A bias term can be added to the result of the propagation.

## **Organization**

The neurons are typically organized into multiple layers, especially in deep learning. Neurons of one layer connect only to neurons of the immediately preceding and immediately following layers. The layer that receives external data is the input layer. The layer that produces the ultimate result is the output layer. In between them are zero or more hidden layers. Single layer and unlayered networks are also used. Between two layers, multiple connection patterns are possible. They can be fully connected, with every neuron in one layer connecting to every neuron in the next layer. They can be pooling, where a group of neurons in one layer connect to a single neuron in the next layer, thereby reducing the number of neurons in that layer. Neurons with only such connections form a directed acyclic graph and are known as feed forward networks. Alternatively, networks that allow connections

between neurons in the same or previous layers are known as recurrent networks.

## **Cost function**

While it is possible to define a cost function ad hoc, frequently the choice is determined by the function's desirable properties (such as convexity) or because it arises from the model (e.g. in a probabilistic model the model's posterior probability can be used as an inverse cost).

## **Backpropagation**

Backpropagation is a method used to adjust the connection weights to compensate for each error found during learning. The error amount is effectively divided among the connections. Technically, back prop calculates the gradient of the cost function associated with a given state with respect to the weights. The weight updates can be done via stochastic gradient descent or other methods, such as Extreme Learning Machines, "No-prop" networks, training without backtracking, "weightless" networks and non-connectionist neural networks.

## **4.4 SYSTEM IMPLEMENTATION**

### **APPLICATIONS**

Because of their ability to reproduce and model nonlinear processes, Deep neural networks have found applications in many disciplines. Application areas include system identification and control (vehicle control, trajectory prediction process control, natural resource management), quantum chemistry, general game playing, pattern recognition (radar systems, face identification, signal classification, 3D reconstruction, object recognition and more), sensor data analysis, sequence recognition (gesture, speech, handwritten and printed text recognition medical diagnosis, finance (e.g. automated trading systems), data mining, visualization, machine translation, social network filtering and e-mail spam filtering. DNNs have been used to diagnose several types of cancers and to distinguish highly invasive cancer cell lines from less invasive lines using only cell shape information.

DNNs have been used to accelerate reliability analysis of infrastructures subject to natural disasters and to predict foundation settlements. DNNs have also been used for building

black-box models in geoscience: hydrology, ocean modelling and coastal engineering, and geomorphology.

DNNs have been employed in cybersecurity, with the objective to discriminate between legitimate activities and malicious ones. For example, machine learning has been used for classifying Android malware, for identifying domains belonging to threat actors and for detecting URLs posing a security risk. Research is underway on DNN systems designed for penetration testing, for detecting botnets,<sup>1</sup> credit cards frauds and network intrusions.

DNNs have been proposed as a tool to solve partial differential equations in physics and simulate the properties of many-body open quantum systems. In brain research DNNs have studied short-term behavior of individual neurons, the dynamics of neural circuitry arise from interactions between individual neurons and how behavior can arise from abstract neural modules that represent complete subsystems. Studies considered long-and short-term plasticity of neural systems and their relation to learning and memory from the individual neuron to the system level.



## TIME

As the name suggests Python time module **allows to work with time in Python**. It allows functionality like getting the current time, pausing the Program from executing, etc. So before starting with this module we need to import it.

**The Python time() function retrieves the current time.** The time is represented as the number of seconds since January 1, 1970. This is the point at which UNIX time starts, also called the “epoch.”

## MATH

Sometimes when working with some kind of financial or scientific projects it becomes necessary to implement mathematical calculations in the project. Python provides the **math module** to deal with such calculations. Math module provides functions to deal with both basic operations such as addition(+), subtraction(-), multiplication(\*), division(/) and advance operations like trigonometric, logarithmic, exponential functions.

In this article, we learn about the math module from basics to advance using the help of a huge dataset containing functions explained with the help of good examples.

## Argparse

Argparse is the “**recommended command-line parsing module in the Python standard library.**” It's what you use to get command line arguments into your program.

The standard Python library argparse **used to incorporate the parsing of command line arguments.** Instead of having to manually set variables inside of the code, argparse can be used to add flexibility and reusability to your code by allowing user input values to be parsed and utilized.

## Python

Python an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Van Rossum led the language community until stepping down as leader in July 2018.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of

Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

### **Main article : Python syntax and semantics**

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal

### **Indentation**

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure.<sup>[1]</sup> This feature is also sometimes termed the off-side rule.

### **Statements and control flow**

#### **Python's statements include (among others):**

- The assignment statement (token '=', the equals sign). This operates differently than in traditional imperative programming languages, and this fundamental mechanism (including the nature of Python's version of variables) illuminates many other features of the language. Assignment in C, e.g., `x = 2`, translates to "typed variable name `x` receives a copy of numeric value `2`". The (right-hand) value

is copied into an allocated storage location for which the (left-hand) variable name is the symbolic address. The memory allocated to the variable is large enough (potentially quite large) for the declared type. In the simplest case of Python assignment, using the same example, `x = 2`, translates to "(generic) name x receives a reference to a separate, dynamically allocated object of numeric (int) type of value 2." This is termed binding the name to the object. Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable. Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc. Successive assignments of a common value to multiple names, e.g., `x = 2; y = 2; z = 2` result in allocating storage to (at most) three names and one numeric object, to which all three names are bound. Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However at a given time a name will be bound to some object, which will have a type; thus there is dynamic typing.

- The if statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
- The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The while statement, which executes a block of code as long as its condition is true.

- The try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
- The raise statement, used to raise a specified exception or re-raise a caught exception.
- The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
- The def statement, which defines a function or method.
- The with statement, from Python 2.5 released on September 2006, which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior and replaces a common try/finally idiom.
- The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block.
- The assert statement, used during debugging to check for conditions that ought to apply.
- The yield statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement co routines.
- The import statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using import: import <module name> [as

<alias>] or from <module name> import \* or from <module name>  
import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>],  
....

- The print statement was changed to the print() function in Python 3

## **CONCLUSION**

The findings of this project on gender categorization and age estimation contributions may be applied to real-time application problems. In this work we propose a multi-task learning framework, to simultaneously predict age and gender from face images. In this project, Human Age and Gender in Python environment by using OpenCV.

## REFERENCES

1. Kamal Acharya. School management system project report. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254873.34023165/v1>
2. Kamal Acharya. A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254873.30191075/v1>
3. Kamal Acharya. A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254872.26972790/v1>
4. Kamal Acharya. Web chatting application project report management system. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254871.18588592/v1>
5. Kamal Acharya. RETAIL STORE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254871.14590154/v1>
6. Kamal Acharya. SUPERMARKET MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172252491.19145062/v1>
7. Kamal Acharya. SOCIAL MEDIA MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172252491.11210579/v1>
8. Kamal Acharya. Online music portal management system project report. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172252488.89734698/v1>
9. Kamal Acharya. COLLEGE BUS MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: <https://doi.org/10.22541/au.172245277.70798942/v1>
10. Kamal Acharya. AUTOMOBILE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: <https://doi.org/10.22541/au.172245276.67982593/v1>
11. Kamal Acharya. Ludo management system project report. Authorea. July 31, 2024. DOI: <https://doi.org/10.22541/au.172243999.98091616/v1>
12. Kamal Acharya. Literature online quiz system project report. Authorea. July 31, 2024. DOI: <https://doi.org/10.22541/au.172243825.53562953/v1>
13. Kamal Acharya. Avoid waste management system project. Authorea. July 29, 2024. DOI: <https://doi.org/10.22541/au.172228528.85022205/v1>
14. Kamal Acharya. CHAT APPLICATION THROUGH CLIENT SERVER MANAGEMENT SYSTEM PROJECT. Authorea. July 29, 2024. DOI: <https://doi.org/10.22541/au.172228527.74316529/v1>



15. Acharya, Kamal, *Online Job Portal Management System* (May 5, 2024). Available at SSRN: <https://ssrn.com/abstract=4817534> or <http://dx.doi.org/10.2139/ssrn.4817534>
16. Acharya, Kamal, *Employee leave management system*. (May 7, 2024). Available at SSRN: <https://ssrn.com/abstract=4819626> or <http://dx.doi.org/10.2139/ssrn.4819626>
17. Acharya, Kamal, *Online electricity billing project report*. (May 7, 2024). Available at SSRN: <https://ssrn.com/abstract=4819630> or <http://dx.doi.org/10.2139/ssrn.4819630>
18. Acharya, Kamal, *POLICY MANAGEMENT SYSTEM PROJECT REPORT*. (December 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4831694> or <http://dx.doi.org/10.2139/ssrn.4831694>
19. Acharya, Kamal, *Online job placement system project report*. (January 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4831638> or <http://dx.doi.org/10.2139/ssrn.4831638>
20. Acharya, Kamal, *Software testing for project report*. (May 16, 2023). Available at SSRN: <https://ssrn.com/abstract=4831028> or <http://dx.doi.org/10.2139/ssrn.4831028>
21. Acharya, Kamal, *ONLINE CRIME REPORTING SYSTEM PROJECT*. (August 10, 2022). Available at SSRN: <https://ssrn.com/abstract=4831015> or <http://dx.doi.org/10.2139/ssrn.4831015>
22. Acharya, Kamal, *Burger ordering system project report*. (October 10, 2022). Available at SSRN: <https://ssrn.com/abstract=4832704> or <http://dx.doi.org/10.2139/ssrn.4832704>
23. Acharya, Kamal, *Teachers Record Management System Project Report* (December 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4833821> or <http://dx.doi.org/10.2139/ssrn.4833821>
24. Acharya, Kamal, *Dairy Management System Project Report* (December 20, 2020). Available at SSRN: <https://ssrn.com/abstract=4835231> or <http://dx.doi.org/10.2139/ssrn.4835231>
25. Acharya, Kamal, *Electrical Shop Management System Project* (December 10, 2019). Available at SSRN: <https://ssrn.com/abstract=4835238> or <http://dx.doi.org/10.2139/ssrn.4835238>
26. Acharya, Kamal, *Online book store management system project report*. (February 10, 2020). Available at SSRN: <https://ssrn.com/abstract=4835277> or <http://dx.doi.org/10.2139/ssrn.4835277>
27. Acharya, Kamal, *Paint shop management system project report*. (January 10, 2019). Available at SSRN: <https://ssrn.com/abstract=4835441> or <http://dx.doi.org/10.2139/ssrn.4835441>

28. Acharya, Kamal, *Supermarket billing system project report*. (August 10, 2021). Available at  
SSRN: <https://ssrn.com/abstract=4835474> or <http://dx.doi.org/10.2139/ssrn.4835474>
29. Acharya, Kamal, *Online taxi booking system project report*. (March 10, 2022). Available at  
SSRN: <https://ssrn.com/abstract=4837729> or <http://dx.doi.org/10.2139/ssrn.4837729>
30. Acharya, Kamal, *Online car servicing system project report*. (March 10, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4837832> or <http://dx.doi.org/10.2139/ssrn.4837832>
31. Acharya, Kamal, *School management system project report*. (July 10, 2021). Available at  
SSRN: <https://ssrn.com/abstract=4837837> or <http://dx.doi.org/10.2139/ssrn.4837837>
32. Acharya, Kamal, *Furniture Showroom Management System Project Report* (March 21, 2021). Available at  
SSRN: <https://ssrn.com/abstract=4839422> or <http://dx.doi.org/10.2139/ssrn.4839422>
33. Acharya, Kamal, *Online Vehicle Rental System Project Report* (March 21, 2019). Available at  
SSRN: <https://ssrn.com/abstract=4839429> or <http://dx.doi.org/10.2139/ssrn.4839429>
34. Acharya, Kamal, *Fruit Shop Management System Project Report* (August 10, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4841048> or <http://dx.doi.org/10.2139/ssrn.4841048>
35. Acharya, Kamal, *Hall Booking Management System Project Report* (December 21, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4841055> or <http://dx.doi.org/10.2139/ssrn.4841055>
36. Acharya, Kamal, *Lundry Management System Project Report* (October 21, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4841059> or <http://dx.doi.org/10.2139/ssrn.4841059>
37. Acharya, Kamal, *A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT* (September 25, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4841209> or <http://dx.doi.org/10.2139/ssrn.4841209>
38. Acharya, Kamal, *A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT* (May 25, 2024). Available at  
SSRN: <https://ssrn.com/abstract=4841210> or <http://dx.doi.org/10.2139/ssrn.4841210>
39. Acharya, Kamal, *ONLINE DATING MANAGEMENT SYSTEM PROJECT REPORT*. (April 25, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4842066> or <http://dx.doi.org/10.2139/ssrn.4842066>

40. Acharya, Kamal, *ONLINE RESUME BUILDER MANAGEMENT SYSTEM PROJECT REPORT*. (April 25, 2021). Available at  
SSRN: <https://ssrn.com/abstract=4842071> or <http://dx.doi.org/10.2139/ssrn.4842071>
41. Acharya, Kamal, *TOLL TEX MANAGEMENT SYSTEM PROJECT REPORT* (August 21, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4842082> or <http://dx.doi.org/10.2139/ssrn.4842082>
42. Acharya, Kamal, *Chat Application Through Client Server Management System Project Report* (June 25, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4842761> or <http://dx.doi.org/10.2139/ssrn.4842761>
43. Acharya, Kamal, *Web Chatting Application Management System Project Report* (April 25, 2022). Available at  
SSRN: <https://ssrn.com/abstract=4842771> or <http://dx.doi.org/10.2139/ssrn.4842771>
44. Acharya, Kamal, *Automobile management system project report* (May 25, 2022). Available at  
SSRN: <https://ssrn.com/abstract=4846917> or <http://dx.doi.org/10.2139/ssrn.4846917>
45. Acharya, Kamal, *College bus management system project report* (April 25, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4846920> or <http://dx.doi.org/10.2139/ssrn.4846920>
46. Acharya, Kamal, *Courier management system project report* (May 25, 2023). Available at  
SSRN: <https://ssrn.com/abstract=4846922> or <http://dx.doi.org/10.2139/ssrn.4846922>
47. Acharya, Kamal, *Event management system project report* (April 25, 2021). Available at  
SSRN: <https://ssrn.com/abstract=4846927> or <http://dx.doi.org/10.2139/ssrn.4846927>
48. Acharya, Kamal, *Library management system project report II* (May 25, 2020). Available at  
SSRN: <https://ssrn.com/abstract=4848857> or <http://dx.doi.org/10.2139/ssrn.4848857>